EXPERIMENTAL STUDY OF THE SCALE TRANSFORM BASED FEATURES IN CONTINUOUS DIGIT RECOGNITION

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by
YOGESH KR. KANDPAL

to the
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
JULY, 2000

Certificate

This is to certify that the work contained in the thesis entitled, "**Experimental Study of the Scale Transform based Features in Continuous Digit Recognition**", has been carried out by **Yogesh Kumar Kandpal** under my supervision, and it has not been submitted elsewhere for a degree.

Dr. S. Umesh
Asst. Professor
Deptt. of Electrical Engg.
Indian Institute of Technology
Kanpur

July, 2000

Acknowledgements

Let me take this opportunity to express deep gratitude towards my thesis supervisor Dr. S. Umesh for giving me tremendous amount of support and time throughout this thesis work. I thank him for providing me lots of motivation and much needed words of encouragement during some moments of despair. I would also like to thank all my teachers at IITK, whose courses helped me in some way or other during my thesis work.

I wish to specially thank Rohit Sinha for providing me the knowledge and help throughout this thesis. Also I wish to thank Shafi and Vinay for their support and excellant company in lab.

I wish to thank my family members for being supportive in my every endeavour. I wish to thank all my friends at IITK for giving me an excellant company during my stay. Finally I thank the Almighty for making it all possible.

Yogesh Kr. Kandpal
July 2000

Abstract

We have studied the use of scale transform based cepstrum as an alternative to widely used Mel cepstrum in the signal processing front end of speaker-independent speech recognition systems. Speaker-independent recognition systems are systems that are trained to recognize speech from many speakers and are, therefore, useful in applications such as telephone based railway enquiry or directory assistance. There is a large difference in performance between speaker-dependent and speaker-independent systems for the same recognition task. This degradation in performance of speaker-independent system is largely due to the variability introduced by inter-speaker variations. This variation among speakers occurs mainly due to differences in vocal tract lengths. It is a commonly held assumption that such differences in vocal tract lengths can be approximated by a linear scaling of the frequency axis. One of the fundamental properties of the scale transform is that its magnitude is invariant to linear scalings in frequency domain and may, therefore, be useful as an acoustic features in speech.

In this thesis we do an experimental study of the application of scale transform to improve the performance of speaker independent continuous digit recognition. The digit recognizer uses a continuous density Hidden Markov Models based system and is implemented using the development environment provided by a toolkit obtained from Oregon Graduate Institute. In the first set of experiments, we compare the performance of Scale Transform based Cepstral Coefficients (STCC) and the Mel Filter bank based Cepstral Coefficients (MFCC). This is done by simply replacing the MFCC features with STCC features for the digit recognition task. The performance of STCC is much lower than MFCC. One possible reason for this degradation is that the STCC features are correlated, and therefore, may not be modeled accurately with a mixture of Gaussian densities with diagonal covariance matrices that are used by the HMM based system. In the second set of experiments, we describe simple methods to approximately decorrelate the STCC feature so that it can be accurately modeled in the HMM based system having diagonal covariance matrices. We show that by using decorrelated STCC features we can obtain a performance that is close to MFCC. This suggests that with a more appropriate model parameterization (i.e. using HMM models with full covariance matrices) the performance of STCC can be significantly improved and may therefore be a robust and practical alternative to MFCC.

Contents

List of Figures

Chapter 1

Introduction

Automatic recognition of speech by machines has a long history of being one of the difficult problems in Artificial Intelligence and Computer Science. By automatic speech recognition we mean a system which takes, as input, the acoustic waveform produced by a speaker and produces, as output, a sequence of linguistic words corresponding to the input utterance. Once such a system is developed it will replace keyboard of computers and the number dialing system of mobiles/phones, and can be found useful in many other applications.

Research in Automatic Speech Recognition (ASR) by machine has been done for almost 5 decades. The earliest attempts to devise systems for ASR by machine were made in the 1950's, when the researchers tried to exploit the fundamental ideas of acoustic-phonetics The problems of segmentation, classification and pattern matching were explored in sixties. In the seventies dynamic programming methods were successfully applied and the area of isolated word or discrete utterance recognition became viable. Speech research in the 1980's was characterized by a shift in technology from template-based approaches to statistical modeling methods especially the Hidden Markov Model (HMM) approach [7].

The following two approaches are widely used in speech recognition.

1. **The pattern recognition approach** - It consists of following steps :

Feature measurement - It converts the speech into some type of parametric representation to define a pattern.

Pattern training - In this step patterns corresponding to speech sounds of same class are used to create a reference pattern for that class.

Pattern Recognition - Unknown test pattern is compared with each of the reference patterns and a measure of similarity is used to decide the class of pattern.

2. **Hidden Markov Model (HMM) approach** - This assumes that the speech signal can be well characterized as a parametric random process, and that the parameters of the stochastic process can be determined or estimated in a precise, well defined manner [3, 7]. Similar to the pattern recognition approach we have following steps :

Feature measurement - It converts the speech utterance into some type of parametric representation.

Model Training - We estimate the parameters of the statistical model for each basic speech unit from a training set of utterance.

Recognition - Scoring of the test utterance with the statistical model and choosing the model with the highest probability for the given test utterance.

At present, we have acceptable speaker dependent speech recognition systems and research is now focused on the development of speaker independent systems. By speaker independent speech recognition systems, we mean, a system which is quite robust to inter-speaker variations. Such a speaker independent system is necessary if the speech recognizer is to be used in applications such as telephone based railway enquiry or directory assistance which may be accessed by many peo-

ple instead of a single person using it. There is degradation in performance when we move from speaker dependent to speaker independent systems. The degradation in performance of speaker independent system is mainly due to the differences in the vocal tract size among the individual speakers which contribute largely to the variabilty of speech waveforms for the same utterance [5]. The differences in vocal tract length manifests themselves as frequency-scaling of the spectral envelopes of speech from different speakers.

The obvious idea is to go for a representation which is invariant to such frequency scaling. The Scale transform is one such representation which has the important property that the magnitude of the scale transform of a function and its scaled version are the same [1]. The use of Scale transform based cepstrum therefore, should help in removing speaker variability due to differences in vocal tract length [9, 8]. Thus the motivation for use of scale transform based features in speech recognizers.

The scope of this thesis is to do experimental study of scale transform based features in a continuous digit recognizer. The performance of a recognizer based on this feature is then evaluated by comparing with mel-scale based features. In this thesis we have given the description of Mel and Scale signal processing front-ends in chapter 2. In Chapter 3 we have given a brief background information of HMM's and the OGI Toolkit. All the results and implementations are discussed in chapter 4. Finally the summary of our experimental results and the conclusion are made in chapter 5.

Chapter 2

Mel and Scale Features

The various speech recognizer developed till date can be classified into different classes depending on the varying emphasis to different algorithmic inputs to this from a wide variety of disciplines, including statistical pattern recognition, communication theory, signal processing, combinatorial mathematics, and linguistics, among others. But the most common denominator of all recognition systems is the signal processing front-end, which converts the speech waveform to some type of parametric representation (generally at a considerably lower information rate) for further analysis and processing. A wide range of techniques exists for parametrically representing the speech signal. These include the short time energy, zero crossing rates, level crossing rates, and other related parameters. One of the important parametric representation of speech is the short time spectral envelope. Spectral analysis methods are therefore generally considered as the core of the signal-processing front-end in a speech recognition system. There are two dominant methods of spectral analysis namely, the bank of filter spectrum analysis model (Mel-scale), and the linear predictive coding (LPC) spectral analysis model. Most modern day speech recognizers use the Mel-scale based spectral analysis. The Scale transform based front end processor has been proposed as an alternative to both Mel and LPC mod-

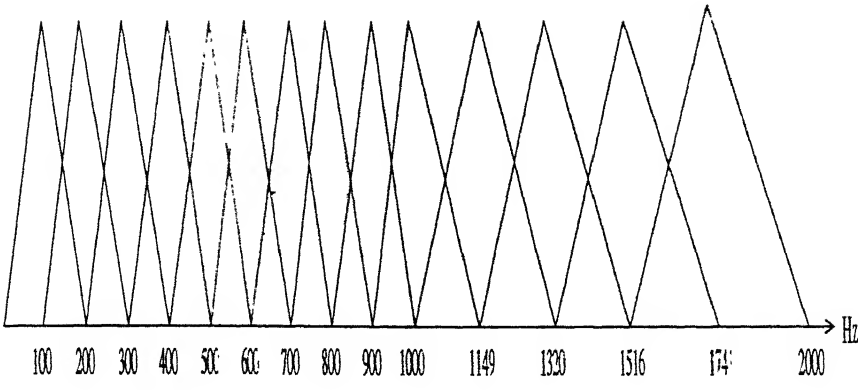

Figure 2.1: Triangular filters placed according to Mel frequency scale

els. We now discuss in detail the Mel and Scale transform based front end processors.

2.1 Mel-scale Filter Bank Front End Processor

The Mel-scale based feature is motivated by perceptual properties of human ear. The response of the human ear to the frequency components in the audio spectrum is non-linear. Differences in frequencies at the low end of the spectrum (< 1Khz) are more detectable than differences of the same magnitude in the high end of the audible spectrum. Filter-bank analysis simulates this type of processing by creating a set of filters in frequency bands spaced in a similarly non-linear fashion [2]. The filter banks have higher bandwidth at high frequencies compared to the filters at lower frequencies. This non-linear relationship is described by the Mel-scale, which relates the physical frequency in Hz to the perceived frequency measured in mels. The spacing of the filters follow the Mel-scale, which is given below.

$$F_{mel} = 2595 \log_{10}(1 + \frac{f}{700}) \tag{2.1}$$

Since the convolutional process of filtering the input time signal is multi-

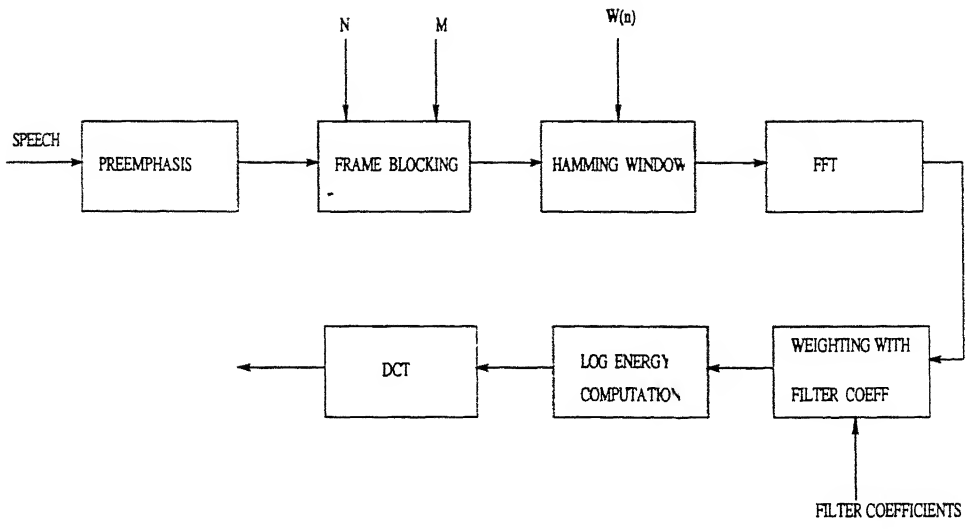

Figure 2.2: Block diagram of a Mel-scale filter bank feature processor

plicative in the frequency domain, the filters are implemented in the frequency rather than time domain. The shape of these filters are triangular. The frequency domain filtering can be viewed as creating a set of bins spaced non-linearly across the frequency spectrum. The value associated with each bin corresponds to the weighted average of the power spectral values in the particular frequency range specified by the shape of the filter. Since the shape of the spectrum imposed by the vocal tract is smooth, energy levels in adjacent bands tend to be correlated. The cosine transform converts the set of log energies to a set of cepstral coefficients, which are largely uncorrelated. Having the features uncorrelated, makes it easier to compute probability estimates in a subsequent statistical modeling. The cosine transform is given by

$$c_i = \sum_{j=1}^{N} m_j cos(\frac{\pi i}{N}(j - 0.5)) \quad where, 0 \leq i \leq N \tag{2.2}$$

In the above equation N represents the number of filter-bank channel selected, m_j the log filter-bank energy values and c_i the resulting cepstral coefficients.
The basic steps required in this feature processing are as follows [7]:

1.**Preemphasis:** The digitised speech signal,$s(n)$,is put through a low-order digital system. to spectrally flatten the signal i.e., it accentuates the high frequency components The most widely used preemphasis network is the fixed first-order system, $H(z) = 1 - \tilde{a}z^{-1}, 0.9 \leq \tilde{a} \leq 1.0$

2.**Frame Blocking:** In this step the preemphasized speech signal, $\tilde{s}(n)$, is blocked into frames of N samples, with adjacent frames being separated by M samples. When $M \leq N$, then the adjacent frames will overlap, and the resulting spectral estimates will be correlated from frame to frame; if $M \leq N$, then spectral estimates will be quite smooth. On the other hand, if $M > N$, there will be no overlap between adjacent frames; in fact some of the speech will be totally lost and the correlation between the resulting spectral estimates of adjacent frames will contain a noisy component whose magnitude will increase as M increases.

3.**Windowing:** The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and the end of each frame by tapering the signal to zero at the beginning and at the end of each frame. If we define the window as $w(n)$, $0 \leq n \leq N-1$, then the result of windowing the l^{th} frame is the signal

$$\tilde{x}_l(n) = x_l(n)w(n), \quad 0 \leq n \leq N-1 \tag{2.3}$$

A typical window used is the hamming window, which has the form

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1 \tag{2.4}$$

4.**Calculation of Energy Vector:** Each of the windowed waveform segments is

transformed into the frequency domain by computing the FFT of the corresponding waveform. A vector of log energies is then computed from each waveform segment by weighting the FFT coefficients by the magnitude frequency response of the filter bank. The log energies are taken for the purpose of dynamic range compression and also in order to make the statistics of the estimated speech power spectrum approximately Gaussian.

5.**Computing DCT:** The final processing stage is to apply the discrete cosine transform (DCT) to the log energy coefficients. This has the effect of compressing the spectral information into the lower order coefficients, and it also decorrelates them to allow the subsequent statistical modeling to use diagonal covariance matrices.

2.2 Scale Transform Based Front End Processor

The scale transform based front end processor has been recently proposed as an alternative to Mel-scale filter bank front end processor. The scale transform based cepstrum is motivated by speaker normalisation techniques [8]. Such normalization techniques are necessary, since different speakers have different formant frequencies for the same vowel.A main source for this variability among different speakers is due to the differences in vocal-tract lengths [11]. If vocal tract is viewed as a uniform tube of length L then the frequency spectrum is given by

$$F_n = \frac{(2n+1)c}{4L} \tag{2.5}$$

where c is the velocity of sound. The length L depends upon the speaker. A popular procedure for normalization is based on the assumption that the formant values of any given speaker are approximately a multiplicative scale factor times the formant values of any other speaker for a given vowel [9]. In other words, the i_{th} formant

frequency of two speakers, A and B for any vowel are related by

$$F_i^{(A)} = \alpha_{AB} F_i^{(B)} \tag{2.6}$$

where α_{AB} is the scale factor. As will be shown in the next subsection the Scale transform is invariant to such scaling.

2.2.1 The scale Transform

Scale is a physical attribute of a signal just like frequency. In the frequency case, we determine the frequency content via the fourier transform; for scale we need a transform which indicates the moment of scale in the signal [1]. The scale-transform of a function, $X(f)$, is given by

$$D_x(c) = \int_0^\infty X(f) \frac{e^{-j2\pi c \ln f}}{\sqrt{f}} df \tag{2.7}$$

and the inverse scale transform is

$$X(f) = \int_{-\infty}^\infty D_x(c) \frac{e^{j2\pi c \ln f}}{\sqrt{f}} dc \tag{2.8}$$

Now the basic property of the scale-transform is that the magnitude of the scale transform of a function, $X(f)$ and its normalized scaled version, $\sqrt{\alpha}X(\alpha f)$, are equal.(Note that $0 < \alpha < 1$ corresponds to dilation, while $1 < \alpha < \infty$ corresponds to compression.) To show this consider the scale transform of $\sqrt{\alpha}X(\alpha f)$, i.e.,

$$D_X^\alpha(c) = \int_0^\infty \sqrt{\alpha} X(\alpha f) \frac{e^{-j2\pi c \ln f}}{\sqrt{f}} df \tag{2.9}$$

Using the substitution of variables, $f' = \alpha f$, we have

$$D_X^\alpha(c) = e^{j2\pi c \ln \alpha} \int_0^\infty X(f') \frac{e^{-j2\pi c \ln f'}}{\sqrt{f'}} df'$$

$$= e^{j2\pi c \ln \alpha} D_X(c) \tag{2.10}$$

Hence, the magnitude of the scale transform of $X(f)$ and its scaled version are the same. The scaling constant α is a part of the phase expression and does not appear in the magnitude of the scale transform. Thus if one were to compute the magnitude of the scale transform of the formant envelope, then all speaker-dependent scaling constant that appear in the phase term would be removed. The scale transform may also be computed as the Fourier transform of the function $X(e^f)e^{f/2}$, i.e.

$$D_x(c) = \int_{-\infty}^{\infty} X(e^f)e^{f/2}e^{-j2\pi cf}df \tag{2.11}$$

It may be noted that as a result of log warping, i.e., forming $X(e^f)$, the speaker specific scale constant, α, is purely a function of the translation parameter in the log warped domain. This can be seen by considering

$$X_1(f) = X(e^f) \tag{2.12}$$

$$X_2(f) = X(\alpha e^f) = X(e^{f+\log\alpha}) = X_1(f + \log\alpha) \tag{2.13}$$

Therefore, if there are two formant envelopes that are related by a pure scaling constant, that is independent of frequency but is dependent on the pair of speakers, then in the log warped domain, the envelopes are the same except for a translation factor dependent on α.

2.2.2 Computation of Scale Transform Based Features

The various steps involved are as following:

1.**Preemphasis and Frame Blocking :** The first two operations are the same as in the Mel-scale processor and has been explained in details in the previous section.

2.**Estimation of the Formant Envelope :** According to the source-filter model for speech production vowels are produced by the vocal tract filter driven by the

source excitation. In the spectral domain this corresponds to the product of the spectrum of the vocal tract filter and the spectrum of the pitch, i.e.,

$$V(f) = F(f)P(f) \tag{2.14}$$

where, V(f), F(f) and P(f) are the observed spectrum, the frequency response of the vocal tract and the spectrum of the pitch excitation respectively. Since we are interested only in the vocal tract response, we would like to remove the effects of pitch excitation. We use the method proposed in [6] to suppress the effects of the pitch. Using this method the frame of speech is segmented in to the 6 overlapping subframes and each subframe is Hanning windowed. We have chosen the subframes of 64 samples and the overlap between them of 45 samples. We estimate the sample autocorrelation for each subframe and average over the available 6 subframes. This averaged autocorrelation estimate is then Hanning windowed and is used to compute the scale cepstrum. We denote the windowed average autocorrelation estimate as $s(n)$. In this method, pitch is effectively suppressed since the duration of each subframe is less than the expected pitch-interval. For every subframe that contains an individual pitch-pulse there is a broadband energy contribution to the spectrum of that subframe and not to any other subframe. The result is that the averaged spectrum contains all of the formant structure but almost none of the pitch structure.

3.**Computing the Scale Cepstrum :** The scale cepstrum is obtained by computing the scale transform of $\log|S(f)|$ and is denoted by $D_s(c)$ where $S(f)$ is the fourier transform of $s(n)$, the windowed averaged autocorrelation estimate. In the calculation of the scale cepstrum, the analytic spectrum is used rather than the symmetric spectrum. Since the scale properties are not valid for symmetric log/mel warped spectrum. The reason for using the logarithm operation, is that it provides a more parsimonious representation in the scale cepstral domain .The logarithm operation affects only the magnitude of the spectral components. Therefore, formant

envelopes that are frequency scaled versions of each other continue to remain so even after the logarithm is taken.The magnitude of scale-cepstrum is then used as a feature vector.

2.2.3 Discrete Implementation of Scale Cepstrum

Since the sampling frequency of the database we wre using is 8KHz, for computation, we assume that the signal is bandlimited between 300-3800 Hz. The scale-cepstrum may therefore be represented as

$$D_S(c) = \int_{300}^{3800} \log|S(f)| \frac{e^{-j2\pi c \ln f}}{\sqrt{f}} df \tag{2.15}$$

Using the substitution of the variables, $\nu = \ln f$, we have

$$D_S(c) = \int_{300}^{3800} \log|S(e^{\nu})| e^{\nu/2} e^{-j2\pi c\nu} d\nu \tag{2.16}$$

which is the conventional fourier transform of $(\log|S(e^{\nu})|)e^{\nu/2}$. For digital implementation [9], we sample in the ν domain and obtain an expression which can be easily implemented using the fast fourier transform (FFT), i.e.,

$$D_S(\left[\frac{k_c C_p}{N}\right]) = \sum_{m=0}^{127} (\log|S(e^{m\Delta\nu+\ln(300)})|) e^{(m\Delta\nu+\ln(300))/2} e^{-j2\pi(k_c/N)m} \tag{2.17}$$

where, $k_c = 0, 1, ..., 127$ and $N = 128$, as we take 2*64 point inverse DFT.
$\Delta\nu = (ln(3800) - ln(300))/(K-1)$ and $C_p = 1/\Delta\nu$.
The phase term $\exp(-j2\pi k_c C_p \ln(300)/N)$ can be ignored, since it does not contribute to the magnitude of $|D_S[k_c C_p/N]|$.

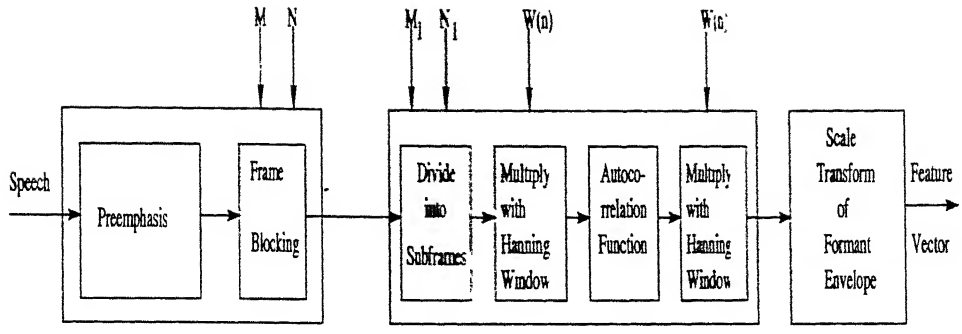

Figure 2.3: Block diagram of a scale transform based front end processor

$S(e^{m\Delta\nu+\ln(300)})$ can be easily computed from the time-lag samples of the formant-envelope, $s(n)$ as

$$S(e^{m\Delta\nu+\ln(300)}) = \sum_{n=-63}^{63} s(n)e^{-j2\pi e^{(m\Delta\nu+\ln(300))}nT_s} \tag{2.18}$$

where, $m = 0, 1, ..., 63$, as we take 64 equally spaced samples in log warped spectrum. T_s is the sampling period in the time lag domain.

The magnitude of scale cepstral coefficients, i.e., $|D_S[k_c C_p/N]|$, are used as features.

Computation of Energy

As Mel-scale based features use energy in place of the first cepstral coefficient, so we have replaced the first cepstral coefficient c_0 of Scale-based features with energy. The energy is computed for each frame after preemphasis. The preemphasized frame is multiplied by a hanning window. The samples of a frame are then squared and added to give the estimate of an energy. We add a small constant of 10^{-6} to each frame energy so that after taking logarithm (as we are computing the energy in decibels), the energy of frames having zero value samples doesn't become infinity.

Chapter 3

Hidden Markov Models and OGI Toolkit

This chapter discusses the Hidden Markov Model (HMM) and the OGI toolkit, which is used for recognition. HMM is a widely used statistical method of characterizing the spectral properties of the frames of speech. The underlying assumption of the HMM is that the speech signal can be well characterized as a parametric random process, and that the parameters of the stochastic process can be determined (estimated) in a precise, well defined manner.

3.1 Definition of HMM

The Hidden Markov Model is a finite set of states, each of which is associated with (generally multidimensional) probability distribution. Transitions among states are governed by the set of probabilities called transition probabilities. In a particular state the outcome or observation can be generated, according to the associated probability distribution. It is only the outcome, not the state which is visible to an external observer; hence the name Hidden Markov Model (HMM).

In order to define an HMM completely, following elements are needed.

1. The number of states of the model, N.

2. M, the number of distinct observation symbols per state.

3. A set of state transition probabilities, $\Lambda = a_{ij}$.

$$a_{ij} = P(q_{t+1} = j | q_t = i), \quad 1 \leq i, j \leq N \tag{3.1}$$

where q_t denotes the current state. Transition probabilities should satisfy the normal stochastic constraints,

$$a_{ij} \geq 0, \quad 1 \leq i, j \leq N$$

and

$$\sum_{j=1}^{N} a_{ij} = 1, \quad 1 \leq i \leq N$$

4. The observation symbol probability distribution in each state, $B = b_j(k)$,

$$b_j(k) = P(o_t = v_k | q_t = j), \quad 1 \leq j \leq N, 1 \leq k \leq M \tag{3.2}$$

where v_k denotes the k^{th} observation symbol in the alphabet, and o_t the current observation vector. Following stochastic constraints must be satisfied,

$$b_j(k) \geq 0, \quad 1 \leq j \leq N, 1 \leq k \leq M$$

and

$$\sum_{k=1}^{M} b_j(k) = 1, \quad 1 \leq j \leq N$$

If the observation are continuous then we will have to use a continuous probability density function, instead of a set of discrete probabilities. In this case we specify the parameters of the probability density function. Usually the probability density is approximated by a weighted sum of M Gaussian distribution $\aleph$,

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} \aleph(\mu_{jm}, \Sigma_{jm}, o_t) \tag{3.3}$$

where, o is the observation vector being modeled, c_{jm} is the mixture coefficient for the mth mixture in state j, μ_{jm} is the mean vectors and $\sum_{jm}$ is the covariance matrices. c_{jm} should satisfy the stochastic constraints,

$$c_{jm} \geq 0, 1 \leq j \leq N, \quad 1 \leq m \leq M$$

and

$$\sum_{m=1}^{M} c_{jm} = 1, \quad 1 \leq j \leq N$$

5. The initial state distribution, $\pi = (\pi_i)$. where,

$$\pi_i = P(q_1 = i), \quad 1 \leq i \leq N \tag{3.4}$$

Therefore we can use the compact notation

$$\lambda = (\Lambda, B, \pi)$$

to denote an HMM with discrete probability distributions, while

$$\lambda = (\Lambda, c_{jm}, \mu_{jm}, \Sigma_{jm}, \pi)$$

to denote one with continuous densities.

The mathematical details and derivations are omitted here, for more comprehensive description the interested reader is referred to [7, 3].

3.2 HMM in Continuous Digit Recognition

In isolated recognition we use one HMM for each speech unit,which can be a word or a subword. But in continuous recognition, this is not possible because a continuous sequence of speech units, which is usually called a sentence, is to be recognized and hence the number of possible sentences may be quite high even for a small vocabulary.In addition there are two fundamental problems associated with continuous recognition.

1. We do not know the end points of the speech units contained in the sentence.

2. We do not know how many speech units are contained in the sentence.

Because of the above problems, the continuous recognition is more complicated than isolated recognition. However HMMs provide a good frame work for continuous recognition also. In this case we connect the HMMs for each speech units in a sentence to form a large HMM [7]. The transitions between the speech units are derived using the so called language model. In our case we are not using any language model for the digit recognition task. Instead we have used the grammar shown in Figure 3.1 . The basic speech unit in our case is phoneme and we build a HMM model for each of the phoneme. The word pronunciation models for each of the

words defined by the grammar are given in table 3.1 and shows the phonetic units contained in each digit. We build a triphone model from individual monophone models by making use of the grammar and the word pronunciation models. This is described in the following sections.
The steps in the recognition process are given below.

1. Spectral analysis and Parametric transform, in which the speech signal, $s(n)$, is converted to an appropriate spectral representation, e.g., Mel Filter bank Cepstral Coefficients (MFCC), Scale Transform based Cepstral Coefficients (STCC).

2. Connected word Recognition, in which the sequence of spectral vectors (corresponding to each frame) of the unknown (test) connected digit string is matched against the models being used. The output of this process is a set of candidate concatenated models ordered by distance (likelihood, probability) score corresponding to the connected digit string. The most likely sequence of models is choosen from the list of candidate concatenated models. The recognizer output is the connected digit string corresponding to the concatenated models being choosen.

3.3 OGI Toolkit

We have made use of "CSLU Specch Toolkit" which apart from many other utilities for speech processing also provides the development environment for the hidden Markov modelling based speech recognizers. This toolkit may be downloaded from http://www.cse.ogi.edu/CSLU/toolkit.

Defining the task

The grammar used by our continuous digit recognizer is depicted in Figure 3.1. This

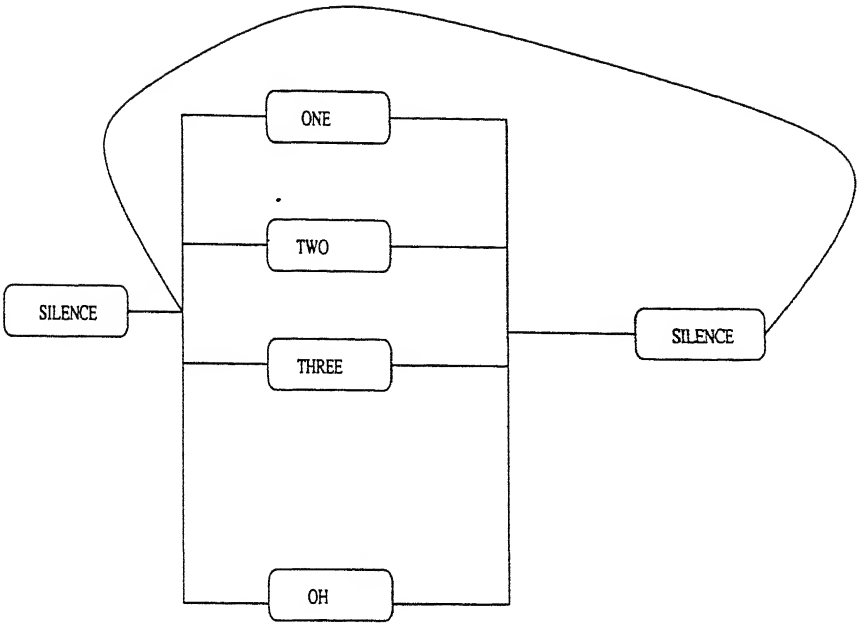

Figure 3.1: Continuous digit string grammar

grammar can be used to recognize any number of spoken digits, with optional silence between words.

Database

We have used the "30k Numbers corpus" obtained from Oregon Graduate Institute (OGI). The numbers corpus is a collection of ordinal cardinal numbers, continuous digit strings and isolated digit strings. The utterances were taken from numerous telephone speech data collection efforts done by Center for Spoken Language Understanding (CSLU) at OGI. This corpus contains 15,000 files. Each file has an orthographic transcription; about 7000 have a phonetic transcription.The speech samples in database are sampled at 8kHz. The corpus is divided into the training set, which is 3/5 of the total data, the development set and the testing set.

Model Prototyping

The word pronunciation models for each of the words defined by the grammar are

one	w ah n
two	t uw
three	th r iy
four	f aor
five	f ay v
six	s ih k s
seven	s eh v ah n
eight	ey t
nine	n ay n
zero	z ih r ow
oh	ow
sil	sil

Table 3.1: Word Pronunciations for the Digit Recognizer

given in table 3.1. These are defined using the ARPABET phonetic alphabet. Based on the pronunciation models we define the first set of HMM models (based on the left to right model) required for the task.

Feature Extraction

We have used two types of features in this work, namely MEL-cepstral coefficients (MFCC) and Scale transform based cepstral coefficients (STCC). The detailed description of their computation is given in chapter 2 (Eqns 2.1 and 2.16) . In both cases along with base features, the first and second order time derivatives were also computed. The time derivatives were computed over 5 frames. Thus the size of feature vector for each frame is 39. We have also done the cepstral mean subtraction in both cases. The cepstral mean subtraction removes the convolutional noise due to channel.

Model Training

The first step to train the chosen mono phone models is to initialize them. The initialization process starts from picking the data segments belonging to particular phone model and loading them into the memory. Each segment is then cut into equal sized segments, depending upon the number of states in the particular model. All data allocated to a particular HMM state is then combined and the initial mixture mean vectors were estimated using vector quantization [4]. The initial mixture variances were set to the pool variance of the data. The parameter estimates are improved using the Viterbi state alignment [7, 3]. During this step state transition probability matrices were not computed. The initial model parameters estimates are further refined using the Expectation maximization (EM) algorithm [7, 3]. In this step initial parameters estimates are also computed for the state transition probability matrix.

Building a triphone model

The finite state search algorithm is designed, using the grammar definition and word pronunciation models to support full cross word triphone modeling. To this end the search build procedure uses a triphone lookup table to determine which model to use during cross word expansion. The left context of the word initials and the right context of the word terminals are assumed to expand to the silence model. The finite state search network is thus created using the word pronunciation models and grammar definition as well as the triphone lookup table.

Transcription

The phonetically hand labeled data are typically only sufficient to create "seed" models for a phoneme based recognizer. To build a more accurate and more robust recognizer requires more data. Most databases contain word level transcription,

which may be used to augment the existing training data. Using the initial model parameter estimates we can create phonetic alignment based solely the word transcription. The input word transcriptions are used to create a finite state grammar where each node or state in the grammar contains a word and its pronunciation variants as they appear in the word pronunciation dictionary. The standard Viterbi algorithm is then used to find the best possible path through the grammar, resulting in the selection of the pronunciation variants to fit the sentence. The resulting output word transcription may be used to generate the associated model transcription for HMM embedded training.

Embedded model reestimation

Until now model training assumed that the phonetic boundaries are defined and there is no interaction between the neighboring models. This problem is addressed in embedded parameter estimation step by creating a composite model by concatenating the models defined by the word transcription file for each of the training utterances specified. Training then continues similar to the the previous training. Using the composite model, however, the results in all models are updated simultaneously.

Evaluation

The performance of the models are now evaluated on previously unseen data, i.e., the data which were not used in the training. The Viterbi decoder searches through the finite state grammar to give recognition answers which are compared with input transcriptions to perform the scoring. All extraneous speech labels are suppressed during the scoring.

Chapter 4

Implementation and Results

In this chapter results of the continuous digit recognition experiments done with Mel Filter bank Cepstral Coefficients (MFCC) and with Scale Transform based Cepstral Coefficients (STCC) are presented. We will show later in this chapter that STCC are correlated whereas MFCC are almost uncorrelated. OGI toolkit used to implement the recognizer supports only diagonal covariances. Therefore we have prewhitened both STCC and MFCC so as to make both of the features uncorrelated. The performance of a recognizer on these prewhitened features are presented in the later half of the chapter. Finally we compare the performance of a recognizer when tested on mismatched data (i.e. training on adult database and testing on children database) for both features.

4.1 Setup Information of the Recognizer

The word pronunciation models (phone models) for each of the words defined by the grammar are given in table 3.1. These are defined using the ARPABET phonetic alphabet. From the pronunciation models we define the HMM models needed for the task. We are using left-to-right HMM models for each of the phone models needed.

Each HMM model consists of 5-states (3 observation states, an entry and exit state). Each state consists of 4 Gaussian mixtures. The initial transition probabilities among states is given below:

0.000 1.000 0.000 0.000 0.000
0.000 0.600 0.400 0.000 0.000
0.000 0.000 0.500 0.500 0.000
0.000 0.000 0.000 0.600 0.400
0.000 0.000 0.000 0.000 0.000

We are doing HMM model initialization by the speech files having the phonetic transcription. We select the 200 utterances of each phone model from the database to initialize the statistics of that HMM model. The transition probabilities get updated during the training of models. The HMM models are embedded trained on training files and then the testing is done on development files. These set of training and testing files are described in the "User's Manual of OGI Toolkit". We are using 39 cepstral coefficients as feature vector for each speech frame which consists of 13 base cepstral coefficients, their first and second order time derivatives.

4.2 MFCC baseline

The built-in feature package in the OGI toolkit is used to compute the MFCC. The 13 base features consists of 12 cepstral coefficients c_1 to c_{12} and an energy in place of c_0. The parameters used for feature computation are as given in Table 4.1.

The result obtained for MFCC is given in Table 4.2. In all tables of result the words, insertions, deletions, substitutions are in actual integer numbers whereas the Word correct, Sentence correct and Accuracy are in percentage. Accuracy will be the figure of merit used in comparing the features tested.

Window size	20ms
Frame size	10ms
Sampling Frequency	8Khz
Preemphasis	0.97
No. of filters	21
Cepstral liftering coeff.	0.6

Table 4.1: Parameters for MFCC

words	3594
insertions	30
deletions	37
substitutions	67
Word correct	97.1062
Sentence correct	87.5444
Accuracy	96.2715

Table 4.2: Recognition performance on MFCC

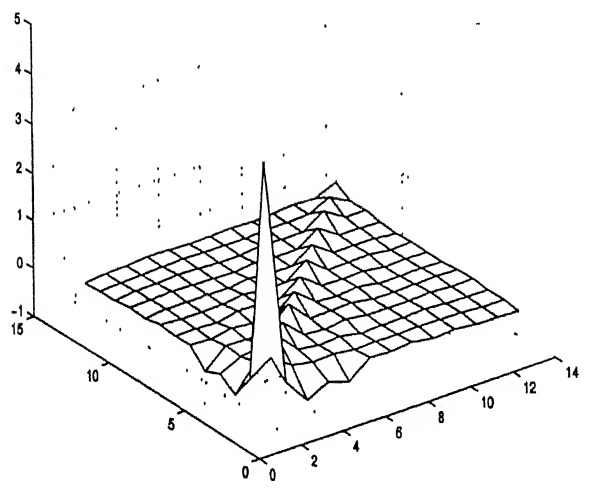

Figure 4.1: Covariance structure of MFCC

The MFCC are almost uncorrelated. The covariance matrix of 13 MFCC are shown in Figure 4.1. 1667821 frames are used to characterize the statistical nature of these coefficients.

4.3 Scale Cepstrum (STCC)

As the toolkit has feature processing module for MFCC only, we have implemented a procedure for STCC computation. We obtain the STCC features for each frame (of 160 samples) of speech by the method proposed by Umesh *et al.* [10] and as described in chapter 2. Each frame of speech is subdivided into six overlapping frames of 64 samples each with an overlap of 45 samples. Each subframe is then hanning windowed. The autocorrelation which is estimated as an inverse FFT of a periodogram estimate is computed for each subframe and then averaged over. This averaged autocorrelation is the smoothed formant envelope estimate, which is mul-

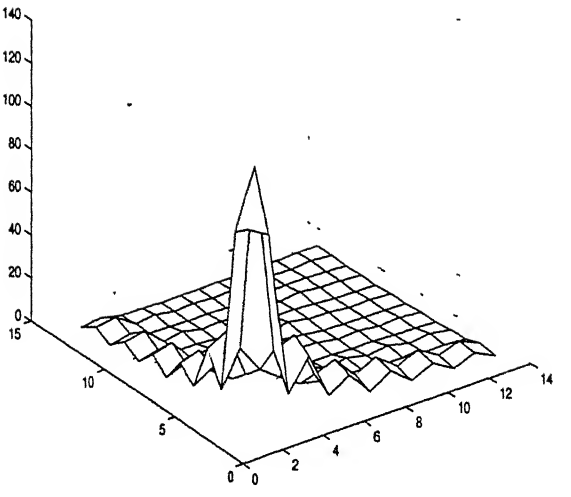

Figure 4.2: Covariance structure of STCC

exponential warping matrix to warp the frequency axis to logarithmic scale. Now we take the logarithm of the resulting vector and compute its inverse FFT to get the cepstral coefficients. We consider the magnitude of first 13 cepstral coefficients from c_0 to c_{12} as our features for a given frame. The parameters used for STCC computation are given in Table 4.3.

The results obtained are given in Table 4.4.

The results show that the performance with STCC is inferior to MFCC for the digit

Window size	20ms
Frame size	10ms
Preemphasis	0.97
Subframe length	64
Subframe overlap	45

Table 4.3: Parameters for STCC

words	3608
insertions	77
deletions	77
substitutions	231
Word correct	91.4634
Sentence correct	67.8486
Accuracy	89.3293

Table 4.4: Recognition performance on STCC

recognition task. The STCC are highly correlated and have the covariance matrix as shown in Figure 4.2 with off diagonal terms of large value. In the figure x-y plane represents the indices of covariance matrix and z represents the correlation value of corresponding indices. Since the STCC features are highly correlated, they may not be adequately modeled by a mixture of Gaussians with diagonal covariance matrices. This may explain the difference in performance.

4.3.1 STCC with Energy

As the recognition in MFCC was done while considering energy as one of the 13 base features, so we give the same advantage to STCC and observe its performance. The energy is computed for each frame by premultiplying with hanning window and then taking square of obtained samples over the frame. The squared samples are added to obtain the estimate of energy over the frame. As the obtained value of energy is very large so we try reduce the dynamic range by computing it in decibels. We add a small constant (10^{-6}) to the energy before converting it in to decibels as there are some speech files in the database having frames of all zeros, which results in infinity on taking logarithm. The obtained value of energy in decibels is used in place of c_0 cepstral coefficient. The parameter used for STCC with energy is the

words	3608
insertions	76
deletions	77
substitutions	220
Word correct	91.7683
Sentence correct	69.1489
Accuracy	89.6618

Table 4.5: Recognition performance on STCC with energy

same as with STCC without energy and is given in Table 4.3. The result obtained for STCC with energy is given in Table 4.5.

There is a marginal improvement in performance as compared to the Scale features without energy .

4.4 Prewhitening of Covariance Matrix

The STCC are correlated and the covariance matrix obtained has offdiagonal terms of large value as shown in Figure 4.2. As the toolkit supports only diagonal covariance so we prewhiten the features to obtain the diagonal covariance matrix. The features can be diagonalized either by Cholesky Decomposition or by Singular Value Decomposition. For Mel the covariance matrix has off diagonal terms of small values, i.e.,the features are not very much correlated as shown in Figure 4.1.

We select the features corresponding to individual phone models and then compute the covariance matrix of each of these models. The global covariance is obtained by taking the average of phone specific covariances. The total number of frames used for estimating the covariance is 53180. We have used two methods to

words	3594
insertions	27
deletions	40
substitutions	78
Word correct	96.7167
Sentence correct	85.8837
Accuracy	95.9655

Table 4.6: Recognition performance on MFCC Prewhitened by Cholesky Decomposition

diagonalize the covariance matrix as decribed below.

4.4.1 By Cholesky Decomposition

In this method, suppose the original feature frame X has the global covariance as C. To obtain the new feature frame having the covariance matrix as identity, we decompose $C^{-1} = D^T D$. On multiplying the original features X by D, new features having an identity covariance matrix is obtained as shown below.

$$E\left[XX^T\right] = C$$

$$E\left[(DX)(DX)^T\right] = DCD^T = D(D^TD)^{-1}D^T = DD^{-1}(D^T)^{-1}D^T = I$$

The results obtained for MFCC and STCC with prewhitening by Cholesky Decomposition are given in Tables 4.6 and 4.7 repectively.

4.4.2 By SVD Decomposition

Singular Value Decomposition (SVD) of inverse of a covariance matrix gives a diagonal matrix S and two unitary matrices U and V. Suppose C is the covariance

words	3594
insertions	25
deletions	88
substitutions	126
Word correct	94.0456
Sentence correct	79.1221
Accuracy	93.35

Table 4.7: Recognition performance on STCC Prewhitened by Cholesky Decomposition

matrix of the speech frames. Now SVD(C)=$[U, S, V]$ and $C = U * S * V^T$ where $V = U$ and S is the diagonal matrix . Then as earlier if X are the old features and $Y = U^{-1}X$ are the features after prewhitening, then we may write

$$E\left[XX^T\right] = C$$

$$E\left[YY^T\right] = U^{-1}C(U^{-1})^T = U^{-1}(USU^T)(U^{-1})^T = S$$

Thus we obtain the matrix U and multiply the features by the inverse of U to obtain the prewhitened features, which have the diagonal covariance matrix S.

The results obtained for MFCC and STCC after prewhitening are given in Tables 4.8 and 4.9 repectively.

Thus Prewhitening is found to affect STCC performance more than the MFCC. This is as expected as the covariance matrix for STCC is more correlated than MFCC. We have also noticed that the results for MFCC degrade by small value on prewhitening. One explaination that strikes to us is the penalty factors that control the deletions and insertions may not be optimum.

words	3594
insertions	25
deletions	35
substitutions	89
Word correct	96.5498
Sentence correct	86.121
Accuracy	95.8542

Table 4.8: Recognition performance on MFCC Prewhitened by SVD Decomposition

words	3594
insertions	26
deletions	64
substitutions	100
Word correct	95.4368
Sentence correct	82.7995
Accuracy	94.7134

Table 4.9: Recognition performance on STCC Prewhitened by SVD Decomposition

words	10207
insertions	181
deletions	520
substitutions	1804
Word correct	77.236
Sentence correct	46.46
Accuracy	75.458

Table 4.10: Recognition performance of MFCC with testing on Preteen data

4.5 Testing with mismatched data

As we mentioned earlier in chapter 3 that Scale transform tries to normalize the variation due to vocal tract length, so Scale transform based features should perform better in mismatched speaker data condition. By mismatch we mean that the database on which the recognizer is trained and the database on which it will be tested are quite different. Previously we trained and tested models with adult speech data. Now we shall evaluate the performance of models trained on adult speech data by testing it on children data. This database consists of speech files collected from children between 10 and 17 years of age and is recorded on the telephone line and has a sampling rate of 8000Hz.

The results obtained for MFCC and STCC are shown in Tables 4.10,4.11 and 4.12.

words	10207
insertions	182
deletions	766
substitutions	1893
Word correct	73.94
Sentence correct	42.97
Accuracy	72.166

Table 4.11: Recognition performance of prewhitened (by SVD) MFCC features with testing on Preteen data

words	10207
insertions	191
deletions	652
substitutions	1904
Word correct	74.9583
Sentence correct	43.36
Accuracy	73.0871

Table 4.12: Recognition performance of STCC with testing on Preteen data

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The following conclusions can be drawn from the experiments done.

1. Using Scale Transform based Cepstral Coefficients (STCC) degrades the performance of a recognizer with respect to Mel Filter Cepstral Coefficients (MFCC) by 6.61 % as shown in tables 4.2 and 4.5 .

2. On analysing the STCC, we found that they are correlated unlike MFCC which are uncorrelated. Their correlation matrix mesh is shown in figure 4.1 and 4.2 . The digit recognizer of OGI is based on the continuous density HMM having diagonal covariance matrices which won't accurately model the correlated features. This explains the degradation observed while using STCC. The results of tables 4.7 and 4.9 confirm our hypothesis by showing that the simple prewhitening of STCC using the global covariance itself improves the performance of a recognizer and matches almost that of MFCC.

3. Motivation for scale is to provide robustness to mismatched speaker conditions. Table 4.10 and 4.12 show results for mismatched conditions. The relative improvement of MFCC over STCC is very small under mismatch conditions which supports the hypothesis that STCC is robust to speaker variations.

We again reiterate that the toolkit we are using supports models having diagonal covariance matrices only. Simple prewhitening of STCC provides performance closer to MFCC. This indicates that with use of proper models with full covariance matrices the performance of STCC may become significantly better than MFCC, which are the current industry standard.

5.2 Scope of Future Work

1. There are many recent modifications in the STCC which may improve the performance of a recognizer [10]. These modifications include the different warping functions other than the logarithmic which we have used.

2. We have assumed the simple global covariance to be the covariance of each phone model. If we shall compute the covariance of each phone model individually and then prewhiten them, it may further improve the performance of a recognizer based on STCC.

3. With use of proper models with full covariance matrices the performance of STCC may become better than MFCC.

References

[1] L. Cohen. The Scale Representation. *IEEE Trans. Signal Processing*, ASSP-41:3275–3292, Dec. 1993.

[2] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monsyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoustic, Speech, Signal Processing*, ASSP-28:357–366, Aug. 1980.

[3] X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models For Speech Recognition.* Edinburgh University Press, 1990.

[4] B. H. Juang and L. R. Rabiner. The segmental k-means algorithm for estimating parameters of hidden markov models. *IEEE Trans. Acoust., Speech, Signal Processing*, 38(9):1639–1641, Sept. 1990.

[5] T. Kamm, G. Andreou, and J. Cohen. Vocal Tract Normalization in Speech Recognition: Compensating for Systematic Speaker Variability. In *Proc. of the 15th Annual Speech Research Symposium*, pages 175–178, Johns Hopkins University, Baltimore, June 1995.

[6] A. H. Nuttall and G. C. Carter. Spectral Estimation using Combined Time and Lag Weighting. *Proceedings of the IEEE*, 70:1115–1125, Sept. 1982.

[7] L. Rabiner and B. H. Juang. *Fundamentals Of Speech Recognition.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[8] S. Umesh, L. Cohen, N. Marinovic, and D. Nelson. Frequency-Warping in Speech. In *Proc. International Conference on Spoken Language Processing*, Philadelphia, USA, 1996.

[9] S. Umesh, L. Cohen, N. Marinovic, and D. Nelson. Scale Transform In Speech Analysis. *IEEE Transactions on Speech and Audio Processing*, January 1999.

[10] S. Umesh, L. Cohen, and D. Nelson. Improved Scale-Analysis in Speech. In *Proc. IEEE International Conference in Acoustics , Speech, and Signal Proc.*, Seattle, USA, May 1998.

[11] H. Wakita. Normalization of vowels by vocal-tract length and its application to vowel identification. *IEEE Trans. Acoustic, Speech, Signal Processing*, ASSP-25(2):183–192, April 1977.